An Interop Publication

CONNEXIONS®

The Interoperability Report

May 1995 Volume 9, No. 5

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.

Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

Subscription hotline: 1-800-575-5717
or +1-502-493-3217

ISSN 0894–5926

From the Editor

As *ConneXions* approaches its 100th issue, I quite often find myself looking back, searching for trends as well as speculating about the future. Our tenth anniversary isn't until May of 1997, so I do have some time to figure out what's been going on, but it never hurts to start thinking about it early.

Looking back, I notice that Open Systems Interconnection (OSI) was a hot topic when this publication began. In April of 1989 we published our first article in a series entitled *Components of OSI.* It was suggested back then that once the series ended we should collect all the articles together, add some "glue" and create a textbook on OSI. Needless to say, this book never materialized. It probably never will given the declining interest in OSI and the incredible success of the Internet suite of protocols or "TCP/IP." This is not to say that OSI is dead, (see this month's Letter to the Editor) nor that you won't see more OSI-related articles in *ConneXions,* it's just a reflection of the fact that things really have changed since 1987.

However, the idea of a book is something I do want to explore further with our more recent series of articles under the heading *Back to Basics.* This time the idea is to explain the various technologies that form the underlying fabric of what is often called "Enterprise Networks." So far we have covered some aspects of network security, the transport layer and electronic mail. This month we take a look at the NetWare protocols IPX, SPX and friends. The article is by Ed Tittel and Dave Smith.

Many of the articles in the *Back to Basics* series result from my own curiosity, or perhaps that inner voice saying "you really ought to know how such-and-such works!" The series is completely open-ended, and I would love to hear your comments regarding what you think we should cover. Please send your suggestions (including pointers to authors) to: `connexions@interop.com.`

If you are tasked with implementing a network for your organization, you will want to read the article about network complexity and its cost by Nick Lippis, John Morency and Eric Hindin.

The NetWorld+Interop 1995 World Tour makes its second stop of the year at the end of the month. This time the venue is Frankfurt, Germany. For more information, check out our World-Wide Web home page at `http://www.interop.com`

Back to Basics: IPX/SPX—The NetWare Protocols

by Ed Tittel and Dave Smith

Introduction

Other articles in this journal have discussed how the US government came up with the TCP/IP protocols to get the maximum benefit from their widely separated and expensive computers. This article takes a look at a more business-oriented approach to some similar issues.

Dawn of an era

Some corporations took to computers from the earliest days of the mainframe. These companies could share files, their groups could work together, and their overall productivity increased markedly. Then, as now, managers generally agreed that more productivity was a Good Thing.

All these users were linked together using the only suitable technology available at the time: Everybody had a dumb terminal on his or her desk, and all these terminals were tied into a monolithic central processing unit called a *mainframe.* This approach wasn't fancy, but it provided to be effective enough to represent a giant leap forward for business computing.

But this was a leap that only giants could make at first. The overwhelming majority of businesses couldn't afford the initial start-up costs of mainframe computing. They read about it. They knew about it. And they were buying time from computer services who sold processing time and services, using somebody else's equipment. Thus, the real benefits of data processing were beyond the immediate grasp of most businesses.

As electronic data processing became more sophisticated and more available, more and more businesses jumped on board. Mainframes got a little smaller and a little easier to use, and the prices started to come down. By the mid 1970s computers were common in businesses of all sizes, except for the smallest "mom and pop" operations.

Enter the PC

When the IBM/PC and its immediate successor, the PC/XT, hit the market in the early 1980s, nearly every business bought one, or two, or a hundred. More and more people in companies had access to computers, or had their own computers. Businesses started to see a dramatic upswing in the quality and the timeliness of their output. At the same time, workers at all levels of business were beginning to feel the power of personal—as opposed to institutional—computing.

Sales divisions created their marketing projections on computers, making them more accurate and therefore more useful to the organization. Salesmen began to keep their contacts electronically. Customer data was organized more efficiently.

Over in accounting, these computers made possible a much wider range of reporting, and dramatically increased the timeliness of that reporting. Personnel departments started keeping employee data electronically, making access to information much more practical, and bringing more science to management. Pretty soon almost everybody had a computer on his or her desk, and the outlook for business seemed brighter than ever.

Trouble in paradigm

But then some of the same problems that the Department of Defense had been trying to solve with the TCP/IP protocols started to crop up in business. All the different departments had been computerized, but all those computers around the company were working in splendid isolation. The information they were creating wasn't being shared throughout the organization because the machines weren't talking to each other.

In a few company divisions, networks were set up. Accounting might have one, but it was a closed system. Only people in accounting had access to that information. The guys in sales got their own network. When the accountants wanted to share some information with sales, and vice versa, they all just got together over lunch.

"Sneakernet" was practiced on a grand scale. Computers had made more information more readily available, but sharing of information between machines, and therefore between different parts of the business, was cumbersome. It didn't take long for the businesses that had jumped onto the computer bandwagon to realize the need for some type of connectivity between all of their machines, and between all of their networks.

We'll send you a copy

By an unremarkable coincidence, others had anticipated that need. Over at Xerox, a company already famous for business automation, work was going on with the stated purpose of setting up business computers to be linked up in networks and to share information. Xerox researchers at the Palo Alto Research Center (PARC) created an architecture for linking business computers called the *Xerox Network System* (XNS), that paved the way for local office networks to be created for businesses. And those networks were to be linked to other networks.

The research and development of the XNS protocols is a story in itself. What's important here is that the XNS architecture, and its associated protocols, were released as an *open standard.* That meant anybody could use and extend this suite of protocols, and several companies released network operating systems inspired by the XNS design.

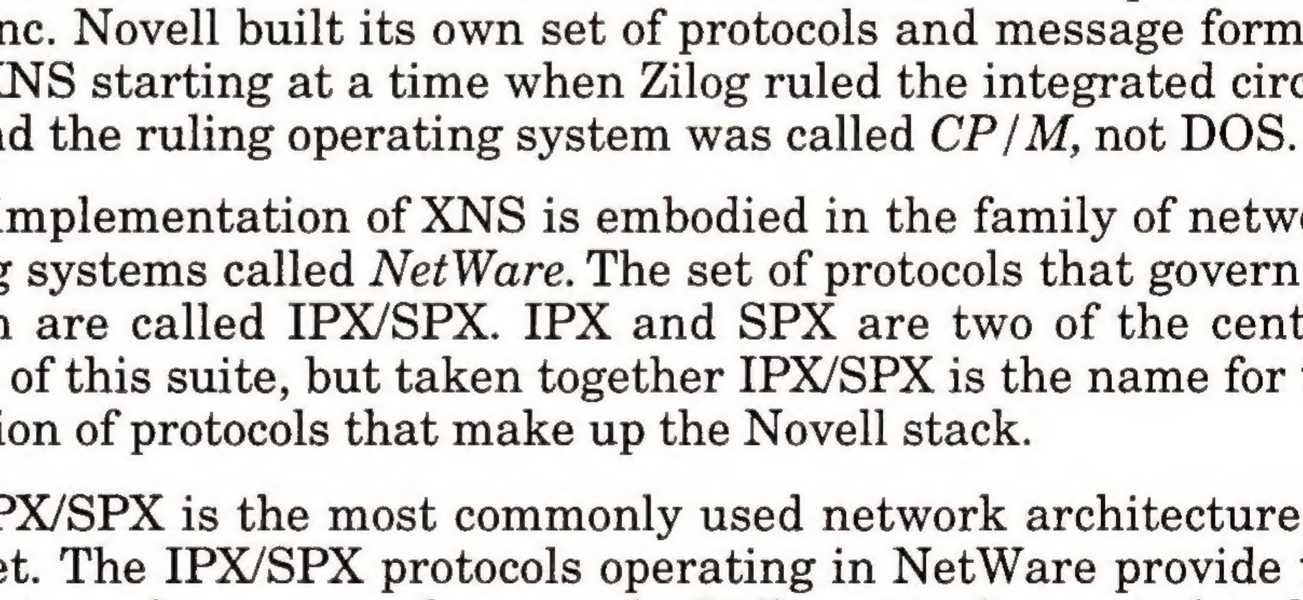
Novell NetWare X.x

Lots of companies created proprietary network protocols based on the XNS architecture. One of the most successful of these companies was Novell, Inc. Novell built its own set of protocols and message formats around XNS starting at a time when Zilog ruled the integrated circuit world, and the ruling operating system was called *CP/M,* not DOS.

Novell's implementation of XNS is embodied in the family of network operating systems called *NetWare.* The set of protocols that govern its operation are called IPX/SPX. IPX and SPX are two of the central protocols of this suite, but taken together IPX/SPX is the name for the aggregation of protocols that make up the Novell stack.

Today, IPX/SPX is the most commonly used network architecture on the planet. The IPX/SPX protocols operating in NetWare provide the same services of any network protocols. Different tasks are isolated on different levels in the stack, and protocols handle the transfer of information between the layers, and across the physical medium.

The following figure is a graphic representation of the IPX/SPX protocol stack:



continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

Although the IPX/SPX architecture stack differs slightly from the *OSI Reference Model,* the layers correspond roughly, and the actual functions of the protocols operating in the layers provide similar definitions and services. It might be helpful at this point to map the Novell architecture to the OSI Reference Model.

<table>
<tr><td>Application</td><td colspan="2" rowspan="3">NCP</td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td><td>NetBIOS (optional)</td></tr>
<tr><td>Transport</td><td colspan="2">SPX</td></tr>
<tr><td>Network</td><td colspan="2">IPX, SAP, RIP</td></tr>
<tr><td rowspan="2">Data Link</td><td colspan="2">ODI driver — Link Support Layer</td></tr>
<tr><td colspan="2">ODI driver — Media Access Layer

Access Protocols (Ethernet, etc.)</td></tr>
<tr><td>Physical</td><td colspan="2">Wiring</td></tr>
</table>

Physical layer

The physical layer is defined by the type of network interface card and the physical medium (type of wire or cable) used to physically connect the machines operating on a network. Network interface cards are usually designed to operate with one particular access method.

Data Link layer

In its proprietary architecture, Novell combines the physical and datalink layers. *Open Data-Link Interface* (ODI) is implemented as a software driver at this layer, and handles the communications between the physical medium and the higher-level protocols in the stack. Following a common approach, Novell chooses to divide the data link layer into two layers so the interaction between the physical layer and the upper level protocols may be more clearly defined. The *Media Access Control* (MAC) sub-layer provides an interface to the physical layer. The *Link Support Layer* (LSL) is a sublayer that provides the interface between the higher level protocols in the stack and the access methods used to get packets onto the physical layer.

Network layer

At this layer Novell provides the *Internet Packet eXchange* protocol (IPX). IPX provides addressing data for network transmissions and provides the delivery system for this data. IPX is a connectionless datagram protocol like IP in the TCP/IP suite. Like IP, IPX is concerned with the addressing and routing of packets. Although it makes its "best effort" to deliver a packet, IPX is not concerned about the connection between hosts, only about addressing and delivery of packets.

The *Services Advertising Protocol* (SAP), running in a NetWare router or a related routing module, broadcasts information about the services offered by servers on the network. A table of services is maintained within the router, and the information in the table is periodically (every 60 seconds) broadcast out onto the internet. Information broadcast by other connected routers is received by SAP and used to update the local table.

The *Routing Information Protocol* (RIP), holds address information about other routers on the internet, and constantly broadcasts and updates this information out onto the internet. Routing tables are maintained on a router, and the information in the tables is broadcast periodically (every 60 seconds) over the internet. RIP receives broadcast information from other connected routers, and uses this information to update its own table.

IPX originally depended on RIP and SAP to find whatever services were available on the network or internetwork and to find the best possible route to the requested network service. Over small and uncomplicated internetworks RIP and SAP perform acceptably, but complicated LANs and larger WANs need more optimal performance. For this reason Novell developed the *NetWare Link State Protocol* (NLSP). NLSP is supported by NetWare version 3.x and later versions. Its purpose is to provide for more efficient use of network resources. Because it is a dynamic update protocol, NLSP helps to reduce the constant traffic fomented by SAP and RIP by broadcasting information only when the network's configuration changes, instead of once every minute.

With the introduction of NetWare 4.x, and the NetWare Directory Services that help to organize and control NetWare 4.x-based (and hybrid) networks, SAP's behavior has also been changed. Because NDS can provide information about all of the services available in a particular NDS directory tree, the only services that have to advertise in such environments are directory services, which can then provide a method to inquire about all other services that might be available. This helps cut down on the "chatty" nature of SAP.

Transport layer

For the transport layer Novell provides *Sequenced Packet eXchange* (SPX). SPX provides the virtual connections between hosts that are necessary for applications to share processing information. Since NetWare is designed for high quality transmission at the datalink layer, and since NCP takes care of some error control and sequencing on the server and the client, SPX is rarely used. It is used primarily for peer-to-peer communications between the NetWare client and server, for utility programs such as RCONSOLE, and PCONSOLE, and for SNA and other gateways.

Session layer

NetWare supports the NetBIOS interface, which provides a link between external network applications and the operating system. Novell's own approach to NetWare communications does not use NetBIOS, however; it is supported primarily for the benefit of applications seeking to take advantage of this service, which is easy for programmers to use to build networked applications. NCP handles session issues for applications, in addition to its application layer role.

Presentation and Application layers

Novell's network architecture, like that of XNS, does not distinguish between the presentation and application layers. The application programs that allow users to interact with the network operating system operate at this level of the NetWare stack, which employ protocols that are used by IPX and SPX to enable network and internetwork communication.

At this layer IPX/SPX provide NCP, the *NetWare Core Protocols*. NCP contains the protocols that define and govern all of the services provided by the NetWare operating system. It is through NCP that clients can have client/server access to a NetWare server, including file, print, directory, chat, and all the other services that NetWare offers.

We'll take a closer look at each of these protocols in the sections that follow. Each of the major Novell protocols will be examined, and we'll see how they work together to provide for reliable network and internetwork communications.

continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

ODI Novell's ODI driver specifications provide a way for a network workstation to utilize different access methods (Token Ring, ARCnet, Ethernet) and their associated protocols and defined frame types while simultaneously providing access to different protocol stacks operating on an internetwork (including protocols like TCP/IP, IPX/SPX, OSI, AppleTalk, etc.).

MLID ODI provides this functionality by coordinating the operations of drivers operating in two sublayers. The *Multiple Link Interface Driver* (MLID) layer is the ODI specification for a network interface card driver that can support all the various access methods in use. The MLID handles communications between all access methods in use on the physical layer and the link support layer. Using an MLID allows higher-level protocols access to any of the various access control methods and the types of physical links that they support. This provides some welcome flexibility on widely-used networks like Ethernet, where a single ODI driver can provide simultaneous support for 802.2, 802.3, and SNAP Ethernet headers without introducing confusion or complications among the separate data streams that are typically involved for each frame type.

LSL The *Link Support Layer* (LSL) switches communications between the access control methods supported by the MLID and the higher-level protocols that operate above the datalink layer. In the case of IPX/SPX, the LSL hands datalink information up to IPX which is operating at the network layer of the protocol stack. If multiple stacks are in use, the LSL can handle switching among the different protocols in use. Here again, Novell's ODI driver technology offers unparalleled support for multiple protocols in simultaneous use (we've regularly run NetWare servers capable of handling all of the following en masse: IPX/SPX, TCP/IP, SNA, OSI, and AppleTalk).

IPX The key protocol in the Novell network architecture is IPX. IPX works together with all the other protocols in the stack, providing a backbone for network and internetwork communication. IPX provides a connectionless, unreliable, datagram service to workstations and servers on an internetwork or a network. Remember that unreliable doesn't mean undependable and lazy in this case, it just means that IPX makes a best-effort attempt to deliver a packet to a destination but requests no acknowledgment to make sure the packet has reached the right spot, or to find out if packets have arrived in the proper sequence.

IPX depends on protocols operating at higher levels in the stack to provide guaranteed delivery, and other services that it does not include. SPX and NCP, both of which we'll cover later in this article, are examples of higher-level protocols in the NetWare suite that are concerned with the quality and reliability of network transmissions.

IPX has two major duties. Its first duty is to format packets. The second is to provide for their delivery. An IPX packet consists of a header (30 bytes) followed by a data section, which can be up to 546 bytes long (in most cases). Because IPX does not provide any facilities for packet fragmentation, IPX implementations must ensure that the packets they send are small enough to be transmitted on any physical networks they want to cross.

IPX requires that all physical links be able to handle IPX packets that are 576 bytes long. (Therefore, the safest approach is to send no packet larger than 576 bytes.)

Many implementations refine this process slightly by detecting when they are sending packets directly to a destination that shares a common segment of networking medium. If it can handle packets larger than 576 bytes (e.g., as with Ethernet, which supports packets up to 1,500 bytes, or Token Ring, which can handle packets of either 4,472 or 17,888 bytes, depending on the particular implementation in use), larger packets can be used.

Formatting IPX packets

Any data that is to be transmitted on a LAN needs to be formatted for transmission over the network. All data handed down to IPX from an upper-layer protocol is encapsulated into an entity known as a packet. IPX is responsible for taking data from the higher levels in the stack and segmenting the data into the packet size determined by the physical transmission media.

Once the packets are cut down to size, IPX adds some information to each packet to ensure proper delivery of the packet to its final destination. This process is most often compared to the way letters are sorted, addressed, and forwarded to their destinations. When you decide to send some type of data through the mail, you find the right size envelope (packet) to put the letter in. On the front of the letter you put its address (destination), and your return address (source), so the receiver knows who sent the letter.

Once all that is done, you're ready to drop the letter in the mail box so it can be forwarded to its destination. This is similar to the process IPX uses to format the data to deliver it to the network.

Anatomy of an IPX packet

An IPX packet contains the following fields:

- *Checksum:* The checksum can be thought of as a fancy parity check. Its objective is to ensure that the bits transmitted are the same bits that are received. In other words, it seeks to ensure that no bits in the packet have been transposed or altered during the transmission. The sending station performs the checksum algorithm on the packet and puts the result in this field. The receiving station will also perform a checksum on the IPX portion of the packet and generate its own checksum. That checksum is checked with the checksum in the packet. If there is a match, the packet is said to be good. If the two do not match, that packet is said to contain an error, and the packet will be discarded. Since this algorithm is performed at the datalink layer in many networking technologies, Novell has opted to disable this feature by default for IPX, considering it to be redundant and time-consuming—with unnecessary cost. It can be enabled, however, as a configuration option.
- *Length:* This field is used to indicate the total length of the IPX packet, including the IPX header checksum field. This means the length of the IPX header and the entire data field. The minimum length allowed is 30 bytes (the size of the IPX header) and its maximum number is typically said to be 576. For communications on a LAN, this number may be as high as the transmission medium allows.
- *Transport Control:* This field is initially set to 0 by the sending station. This field counts the number of hops (the number of routers) the packet encountered along the way. Since the maximum number of routers a packet is allowed to traverse is 15 (a network 16 hops away is considered unreachable), the first 4 bits are not used.

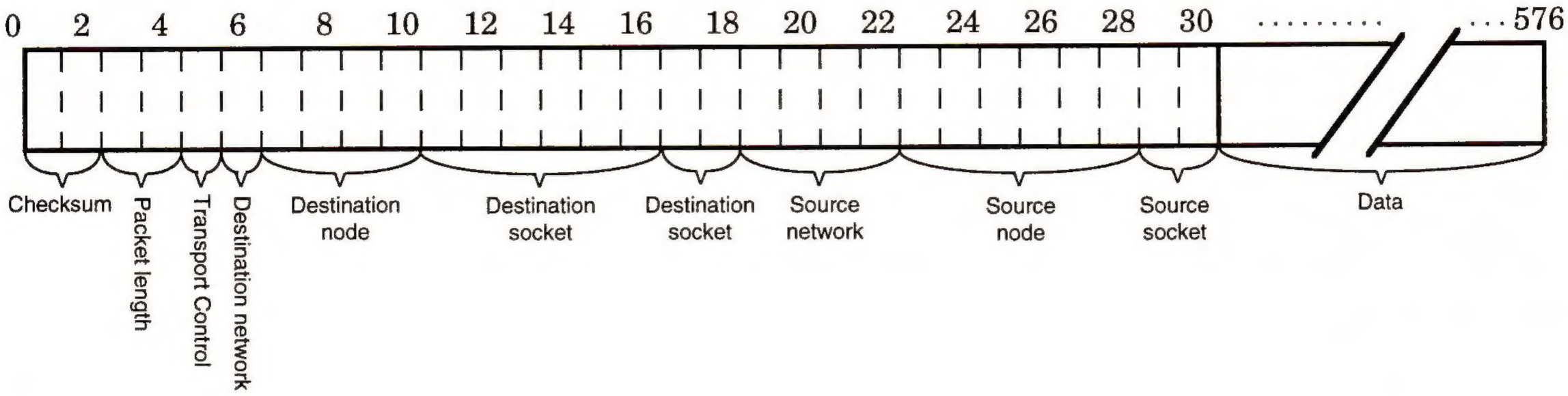
continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

This is also used by routers and other file servers that support Service Access Protocol (SAP) reporting to indicate how far away a server (providing certain services) is from the recipient of the packet. When a packet is transmitted onto the network, the sending station will set this field to 0. As the packet traverses each router on its way to the destination, each router will increment this counter by 1. The router that sets it to 16 also discards the packet.

- *Packet Type:* This field is used to indicate the type of data in the data field. This is the Xerox registration number for Novell NetWare. It identifies the XNS packet as a NetWare packet. Since IPX is a derivative of XNS's IDP protocol, it follows the assigned types given by Xerox.
- *Destination Network:* This 32 bit field contains the physical address of the destination network on which the destination host resides. An analogy is that a network number is like the area code of the phone system. It is used by IPX in routers and workstations to determine if the destination host resides on the local network or a different network.
- *Destination Host:* This 48-bit field contains the physical address of the final (not any intermediate hosts) destination network station. An analogy of this is the address displayed on the letter. Another analogy is the seven-digit number (not including the area code) in the telephone system.
- *Destination Socket:* This 16-bit field is an indicator of the process to be accessed on the destination station. A socket number is an integer number assigned to a specific process running on a network station. Each and every service that runs on a file server will be assigned a socket number. Any workstation requesting a service must set this field to the proper socket number for the service requested in order to be properly serviced. Some socket numbers are constant and are called *well-known sockets.* Well known socket numbers are assigned to specific applications and services and they never change. Other socket numbers are assigned dynamically.
- *Source Network:* This 32-bit field contains the network number of the source network. This indicates the network number from which the packet originated. A network number of 0 indicates the physical network of the source is currently unknown. If a router receives a packet with no network number, it will assign one. When IPX is initialized, it may obtain the number from the workstation. It may also find its network number from the router. Network numbers are not assigned to the workstation.
- *Source Host:* This 48-bit field contains the physical address of the source host (the network station that submitted the packet). This represents the host number from which the packet originated. Like the destination host field, if the physical address is less than 6 bytes long, the least significant portion of this field is set to the address and the most significant portion of the field is set to 0s Otherwise, it is set to the 48-bit address of the LAN interface card.
- *Source Socket:* This 16 bit field contains the socket number of the process that submitted the packet. This number is usually defined dynamically at the source requester.

The following figure is a representation of the makeup of an IPX packet header:



Sockets explained

Multiple processes may be running on a workstation, and will definitely be running on a file server. *Sockets* are the addresses that indicate the endpoints for communication between like processes. A unique socket number indicates which process running on the network station should receive and transmit data.

Sockets represent an application process running on a network workstation. There are two types of sockets: *static* and *dynamic*. Static sockets are reserved sockets that are assigned by the network protocol or by an application or service. Static sockets are the "property" of a particular service, and cannot be used by any other process on the network. Their socket number never changes. For this reason they are called *well-known socket numbers*. Dynamic sockets are assigned randomly and can be used by any process on the network.

For example, to access the file services of a server, IPX would fill in the destination and source network, the destination host number of the file server, and source host number of its workstation. The destination socket number would be set to 0451 (hex). This is known as a well-known socket number defined by Novell. The source socket (assigned by IPX at the source workstation) will be a dynamic number and IPX will choose an unused address in the range of 4000 to 6000 Hex.

The source socket is used by the destination as a socket number to reply to. It indicates the socket number that made the request. In this way, when the packet arrives at the server, the server will know that the packet is destined for it (the host number) and will also know the transmitting station is requesting something from the server (socket 0451). Deeper into the packet will be an NCP control code that defines exactly what the transmitter of the packet wants to do (create a file, delete a file, directory listing, print a file, etc.).

Once the command is received and decoded, the server will return data to the transmitter of the packet. But it needs to know which endpoint of the workstation will receive this data (which process submitted the request). This is the purpose of the source socket number. The file server will format a packet, reverse the IPX header fields (source and destination headers), set the destination socket number to the number indicated in the received packet of source socket number, and transmit the packet.

continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

Finally, the socket number (source or destination network number, source or destination host number, and source or destination port number) is the absolute address of any process on the network. With the combination of these fields, any process on any network on any network station can be found. IPX controls all socket numbering and processing.

That was one of the functions provided by IPX—the formatting of data into a packet so that it may be transferred across the network. The next function provided by the IPX protocol is routing packets directly to workstations on the same LAN or to a network station on a remote LAN.

IPX routing

The routing function allows packets to be forwarded to different networks through the use of a device known as a *router*. There are two available types of routers on a Novell network, *internal* and *external*.

The internal router is one that is usually performing some other tasks as well as the routing function. These tasks may be file and print services, or a gateway service to SNA. Internal routers usually are concerned with maintaining the flow of data through a LAN. The external router is a PC with more than one network interface card installed. Its sole function is to route packets. External routers are usually used to link one LAN to another to create internetworks. Where Novell used to ship software for external routers with the NetWare OS, it now sells a separate product called the Novell *Multiprotocol Router* (MPR) that competes with routers with similar capabilities from other companies like Cisco Systems, Bay Networks, and others.

In fact, there are a number of companies that manufacture and install routers that are compliant with Novell's IPX/SPX scheme. Usually these routers are multiprotocol routers that will route other types of packets (TCP, AppleTalk, DECnet, etc.) as well as NetWare packets. The protocols are routed simultaneously in the same router.

To route a packet, routers will accept only packets directly addressed to them and will determine the best path on which to forward the packet. Then, they will either forward that packet on to a local workstation or server (if it's destined for delivery on a segment of medium directly accessible to the router), or onto another router (if it's not directly accessible to the present router).

IPX routing tables

Routers need to know of all other available routers and therefore all other active networks on its internet. The IPX router keeps a complete listing of the networks listed by their network numbers. This is known as a *routing table*. Each router in a NetWare internetwork will contain a routing table. The entries in the routing table will let the router determine which is the best path for forwarding packets. Routers on an internetwork exchange information about one another through the Routing Information Protocol (RIP).

A routing table contains a listing of network numbers and an associated path, in order to deliver the packet to its final destination network.

The entries in the routing table do not contain any physical addresses of the network stations that reside on the internet. The only physical addresses in the table are those of other routers to which packets, destined for a remote network, may be addressed. Routers do not know which other end stations are on the networks they connect to.

The final destination (physical address of the final destination) is embedded in the IPX header (the destination host.) Once the router determines that the final destination network number is directly attached to the router, it will extract the destination host number from the IPX header and address the packet and deliver it over the directly attached network segment.

RIP

To exchange their tables with other routers on the internet, IPX uses the *Routing Information Protocol* (RIP). (Do not confuse this RIP with the one used by TCP/IP). RIP is a service residing on the network layer of the stack. Its purpose is to provide IPX with up-to-date address and route information so that packets can be expeditiously forwarded to their destination. The functions of RIP are to allow:

- Workstations to attain the fastest route to a network by broadcasting a route request packet which will be answered by the routing software on the Novell file server or by a router supporting IPX RIP.
- Routers to exchange information or update their internal routing tables.
- Routers to respond to RIP requests from workstations.
- Routers to become aware when a route path has changed.

When an IPX router starts up, it puts the network numbers of the directly connected routers into its routing table. These network numbers are entered by the network administrator when the router is installed. Once integrated, the router will then send a broadcast packet to the network (on each of its directly connected cable segments) containing these routes (the network numbers of the directly attached cable segments) that the router will now make available. Other routers on those cable segments will read this information and update their tables.

The router will then transmit another RIP packet requesting information from other routers on its directly attached network segments (a RIP request). This request will be responded to by any other active routers on the directly connected segments. The term *directly connected segments* is used here because request and response packets are sent in broadcast mode. This means that all stations on the local network will receive and process this packet. These broadcast packets are not forwarded to other networks by the routers. Routers update their tables and, in turn, will broadcast their updated tables to routers that they are directly connected to. The router will then compare the received table to its own table and make any changes necessary.

Once these events have taken place, the IPX router will place itself in the operation of receiving information processing RIP requests, routing packets, and maintaining its own routing table. In addition to these updating tasks, all routers will broadcast their routing tables every 60 seconds.

Determining a local or remote path

When a router is fully operational, other network stations may use that router to forward a packet to those remote networks. Any time a network station wishes to send information to a destination station, it must have the network address as well as the physical (datalink or MAC) address of the destination station. If the two stations are communicating on the same network (sharing the same network number), the transmitting station can send the packet directly to the destination without using a router.

continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

But if the destination station is on a different network the transmitting station must find a router to submit the packet to. The transmitting station depends on that router to find the best route to the destination.

To find a router, the network workstation must transmit a RIP request packet. Inside this packet is the destination address (network number) of the final destination. This request will be answered by routers only on the immediate (same network as the requesting station) network. Routers that are not directly connected to the same network will not see this request because it is broadcast only over the network. Local routers can respond to this type of packet but will not forward this or any direct broadcast packet. Any router that has a path to that destination in its own routing table will respond, and the network workstation will choose the router to forward the packet to. Usually, this will be based on the lowest tick(time) or hop count (number of routers involved in forwarding the transmission).

On receipt of a request, a router sends out a response packet containing its own router's physical address. The requesting network station will use this address to physically address its packet to the router. The router is responsible for finding the location of the destination and for forwarding the packets.

RIP request and response information is encapsulated by IPX for transmission onto the network. These IPX packets, since they don't contain user data or requests, are known as *control information packets,* or overhead. They aren't participating in user data transfer, but they are necessary for the proper functioning of the network.

SAP

Another type of control information is provided by the *Service Advertising Protocol* (SAP), which, like RIP, operates at the network layer of the stack. In order for a workstation to find a server on the network, to login or out of the network, to use print, e-mail, or file services, it must be able to locate a server and the services running on that server. Routers keep tables of server names, their full internet addresses, the services they provide, and their distance from the workstation. The process of maintaining and sharing this information across the network is carried out by SAP.

Routers and file servers maintain tables of service information. SAP is a routing service, but routers update their SAP tables in the same way that they update RIP tables.

When a server is initialized on the network it broadcasts its table of services to the network to let the network know that these services are now available. *Service Identity Packets,* encapsulated in IPX headers, are used to carry this information. They are periodically broadcast over the network, and attached routers and servers update their own SAP tables accordingly. When a server is brought down for any reason it broadcasts that its services are no longer available, and the other routers and servers attached delete the services provided by that server from the SAP tables.

Workstations seeking services send out *Service Query Packets,* encapsulated in IPX headers, to find any active server offering the required service. Routers that receive these packets check their SAP tables and respond by broadcasting Service Identity Packets that contain the full internet addresses of the requested servers. The workstation takes the address information and uses it to address its request directly to the server in an IPX packet.

NLSP

RIP and SAP were both designed to provide a capability for transparent access to servers and routers on an internet. But when a large number of servers and routers are in operation, the overhead from the RIP and SAP broadcasts becomes considerable and has the effect of slowing down network performance, as increased bandwidth is devoted to updating and maintaining information in the RIP and SAP tables.

The *NetWare Link Services Protocol* (NLSP) differs from RIP and SAP in that it is a "link state protocol." Rather than broadcasting service or route information every 60 seconds whether there has been a change in the internetwork or not, NLSP updates the routing or service tables only when a change has been made to the state of the network. NLSP periodically checks the connection to links on the network. If a primary link should go down, NLSP switches automatically to another link so that service is not completely interrupted.

NLSP also removes the limitation of no more than fifteen routers between endpoints on an internetwork. In the early days of networking, this seemed like a logical and prudent limitation. But as networks grew and began to be linked up through routers, the limit of fifteen routers became a problem, and a limitation on internetwork sizes and reach.

NLSP can be a replacement for RIP and SAP, or it can operate in conjunction with the older protocols, allowing it to act as a gateway protocol for separate RIP and SAP domains. Each router using NLSP maintains an Adjacency Database (table), where information about the router's direct links and immediately connected nodes is located; and a Link State Database (table), which is a connection map of the entire internetwork.

Three new types of packets are used by NLSP to carry out its functions:

- Hello packets
- Sequence Number packets
- Link State packets

Through the use of Hello packets, an election process occurs among the routers on an internetwork. One router is elected as the *Designated Router,* and thereafter maintains a pseudonode, or logical representation of the internetwork, in its link state database. Hello packets are also used to send out alerts to neighboring routers whenever a router link has failed for some reason (which might indicate a router failure, or perhaps just a machine being shut down for some reason).

Link State packets are sent out by a router which detects a change in the network topology. The router that finds such a change builds an LSP, and sends it out to its neighboring routers. The LSP does not contain a completely redrawn network map. Only the area affected, which therefore needs to be changed, is transmitted in the LSP.

Sequence Number packets come in two types. Partial Sequence Number packets are used by routers to acknowledge receipt of LSPs. Sequence Number packets contain a list of all areas reported as changed in LSPs, and the sequence of their arrival (sequence number). A router receiving the LSP can compare it to its own table and request the necessary update.

continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

NLSP may be implemented over NetWare 3.x as an add-on NLM, but it is included as part of the standard NetWare 4.x release. Network routers may use either NLSP or RIP/SAP, or they can be used simultaneously. This latter approach is common where older NetWare networks need to be interconnected for intermittent or occasional use; it obviates the need to update every router, yet permits the old 15-hop limitation to be overcome, and reduces the overhead that a pure RIP/SAP routing scheme might otherwise impose.

NetWare Core Protocols

The *NetWare Core Protocols* (NCP) provide the definition for the NetWare operating system, and provide the definition for the services that operating system gives to its users. All of the services contained in NetWare are contained in one kind of NCP packet or another. In this way, NCP defines a language for communication between the servers and clients on an internetwork.

NCP messages are prepared and sent using the formats and conventions specified by Novell's defined NCP standards. NCP is the language NetWare servers and clients speak when they are requesting or delivering services from or to one another.

Clients make requests of servers using the NCP workstation shell, or its *Virtual Loadable Module* (VLM) equivalent. The client transmits messages defined by NCP to request file reads and writes, to create print jobs or monitor print queue status, to determine drive mappings, to search through directories on a file server, and so on.

Servers answer NCP requests by providing access to services that have been defined by NCP. Some examples of such services include:

- *Accounting Services:* keep track of server transactions
- *Bindery Services/Directory Services:* provide access to a database of information about network resources
- *Message Services:* allow messages to be broadcast to servers or clients on a LAN or WAN
- *Print Services:* allow clients on the network to access network printers
- *NetWare Loadable Modules (NLM):* these are external programs that run in a Novell server that are not governed by NCP yet still act like a direct server service. NLMs may be developed by Novell as additions to predefined services, or by third party application developers wishing to extend NetWare's services.

NCP requests and replies are encapsulated in IPX headers for transmission across the network, or internetwork. But the various types of NCPs define most of what makes NetWare the most popular network operating system in use anywhere in the world today.

SPX

The *Sequenced Packet eXchange* (SPX) is the Novell transport layer protocol It was derived from the Xerox Sequenced Packet Protocol (SPP). SPX provides a reliable, connection-oriented, virtual circuit service between network stations. SPX makes use of the IPX datagram service to provide a sequenced data stream. This is accomplished by implementing a system that requires each packet sent to be acknowledged. It also provides flow control between the network stations and ensures that no duplicates are delivered to the remote process.

SPX reduces the number of times that an unneeded retransmission occurs to decrease the congestion on the network. Retransmissions normally occur after the sending station has timed-out waiting for an acknowledgment of a packet that has been lost, damaged, or dropped. SPX uses an algorithm to estimate accurate retransmission times. It also uses historic information to determine the initial time, and it then increases the time by 50 percent if a time-out occurs. The process continues until a maximum time-out value is reached or until acknowledgments return in time and retransmissions are no longer required. In the second case, the time-out stabilizes at a value that is workable for prevailing network conditions.

SPX adds 12 bytes to the IPX packet header, mostly to carry connection control information. Added to the 30 bytes of the IPX header, this results in a combined header of 42 bytes. The maximum size of an SPX packet is the same as the maximum for an IPX packet (which means that the data portion is 12 bytes less, because of the overhead that SPX itself adds).

The developers of NetWare were concerned with reliable and efficient transfer over LANs and WANs. Software was implemented in NCP to provide some of the services of the transport layer, where SPX resides. The NetWare client shell and VLMs also have some simple transport functions included. For this reason SPX is rarely used in Novell networking, except for server-to-server communications and applications that require a special degree of reliability.

In conclusion

This pretty much covers the basics of Novell's IPX/SPX protocol stack. By now you should have a good foundation in the basics of NetWare operations. To summarize, here's a brief rundown of how this stack operates:

- Data is formulated for transmission at the application layer of the stack and passed down to SPX.
- SPX is used to provide a virtually error-free connection between remote applications.
- Data from the higher layers of the stack is sent down to IPX, where the data is segmented into packets of the correct length, and addresses are added to allow network and internetwork transmission.
- IPX attaches the proper header information and hands the packet down to the LSL at the datalink layer.
- The LSL hands the packet down to the MLID, a driver designed to communicate with the various access methods in use for the various physical transmission media.
- The access method defines the way that packets are transported over the intervening physical media used for the network transmission.

One reason IPX/SPX is the most popular protocol suite in use today is that IPX supports nearly every conceivable interface method and physical transmission medium in use in the networking world. For more of the details on IPX/SPX, please consult the resources that follow in the "Annotated Bibliography" section.

continued on next page

IPX/SPX—The NetWare Protocols *(continued)*

Annotated bibliography

[1] Bearnson, Stephen, "Communication Basics and Open Data-Link Interface Technology," *NetWare Application Notes,* Vol. 3, No. 11, November 1992. *Basics of protocol stack operation, lots of ODI.*

[2] Chappell, Laura A., Hakes, Dan E., *Novell's Guide to NetWare LAN Analysis,* 2nd Edition, 1994, Sybex, Inc., Alameda, CA. *When you want the real thing, go right to the source!*

[3] Miller, Mark A., *LAN Protocol Handbook,* 1990, M&T Publishing, Inc., Redwood City, CA. *Good technical reference for those already in the know.*

[4] Mosbarger, Myron: "A Review of Bridging and Routing Technologies," *Novell Application Notes,* Volume 5, No. 3, March 1994. *What is bridging? Routing? Good background on how networks and internetworks are assembled.*

[5] Naugle, Matthew, *Network Protocol Handbook,* 1994, McGraw-Hill, Inc., New York, NY. *All the important protocol suites are discussed in detail. For a broad range of protocols, this is a good, informative work.*

[6] Tittel, Ed; Deni Connor, and Earl Follis, *NetWare for Dummies,* 2nd Edition, IDG Books, 1993 and 1995. *An informative, lighthearted look at a serious subject, and a good general introduction to networking.*

For further reading

[A] Rose, Marshall, & McCloghrie, Keith, "Back to Basics: The Transport Layer," *ConneXions,* Volume 8, No. 6, June 1994.

[B] Radia Perlman, *Interconnections: Bridges and Routers,* ISBN 0-201-56332-0, Addison-Wesley, Reading, Massachusetts, 1992.

[C] *ConneXions,* Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[D] *ConneXions,* Volume 5, No. 1, January 1991, "Special Issue: Interdomain Routing."

[E] M. Rose, *The Internet Message: Closing the Book with Electronic Mail,* ISBN 0-13-092941-7, Prentice-Hall, 1993.

[F] Crocker, David, "Back to Basics: Internet Electronic Mail," *ConneXions,* Volume 9, No. 1, January 1995.

[G] Chapin, Lyman & Piscitello, David, *Open Systems Networking: TCP/IP and OSI,* ISBN 0-201-56334-7, Addison-Wesley, 1993.

[This article is based on material in *The PC Network Handbook,* by Ed Tittel and Dave Smith, ISBN 0-12-691398-6, due to be published in July 1995 by Academic Press Professional. Used with permission. —Ed.]

ED TITTEL is a full-time freelance writer and networking consultant, and a member of the NetWorld+Interop Program Committee. Ed is the author of ten books on computer topics, many of them network-related. In addition to *NetWare for Dummies,* now in its 2nd edition, Ed is the author of the forthcoming IDG books *HTML for Dummies* and *Foundations of WWW and HTML Programming,* as well as books on the Internet, network design, and e-mail for Academic Press. In his spare time, Ed likes to walk his hefty but personable Labrador, Dusty, and to share the blessings of domesticity with his wife, Suzy, and stepchildren, Austin and Chelsea. His e-mail address is: `ed@etittel.zilker.net`

DAVE SMITH is a network industry consultant and freelance writer, who has labored in obscurity under the corporate umbrella until quite recently. A collaborator with Ed on the forthcoming Academic Press Professional book: *The Network Handbook* (from which this article has been adapted), Dave is also the author of numerous WWW pages and a variety of marketing-oriented high-tech literature. A denizen of Dripping Springs, Texas, Dave enjoys country livin' and his family from the vantage point of a lovely piece of property in the beautiful Hill Country. His e-mail address is: `dbsmith@comland.com`

Letter to the Editor

Dear Ole,

Thank you for reprinting Tim Dixon's article on IPng and OSI in your March 1995 edition. Since we made some progress in this area at the April 1995 IETF meeting, your readers might like an update.

Firstly, the various Internet Drafts that Tim mentioned were replaced by a combined draft with seven co-authors from three continents. Its title was "Mechanisms for OSI NSAPs, CLNP and TP over IPv6" which is quite a mouthful if you expand all the acronyms. Basically it combines all the OSI-to-IP address mapping ideas mentioned in Tim's article in a compatible way. It also gives outline specifications for encapsulating the OSI connectionless datagram protocol inside IPv6 and for carrying OSI transport over an IPv6 network.

Tunneling

We discussed the open issues in this draft during the IETF meeting, and managed to close off most of them. The fragmentation issue Tim raised is quite complex. If CLNP packets are sent through an IPv6 tunnel, then the tunnel could fragment the IPv6 packets if necessary, and the far end could re-assemble them. However, this would fail if the tunnel was actually an "anycast" tunnel in the very latest IPv6 terminology—i.e., a tunnel with multiple end-points, each exiting the IPv6 world and re-entering the CLNP world. A single CLNP packet fragmented into multiple IPv6 packets would never get correctly re-assembled if its fragments arrived at different end-points of the same tunnel. However, this is easily avoided by specifying that if the tunnel's *Maximum Transmission Unit* (MTU) is too small then CLNP *segmentation* (the OSI equivalent of IP fragmentation) must be used instead. Think about it; it works, because the CLNP segments are now small enough to get through before they ever enter the tunnel.

We still had a few issues to solve at the end of the discussion, but this will get done as long as we can find volunteers to push the documents along and maybe even write some code.

Liaison

Jack Houldsworth, liaison from ISO/IEC JTC1/SC6 to the IETF, also gave us an encouraging report on progress inside SC6. Now that the formal agreement between SC6 and the IETF has been signed, SC6 is setting up a specific work plan concerning IP, and has already issued its first related document for CD ballot. This is a formal recognition of Internet Standard 35 (RFC 1006) which allows OSI transport to run over TCP.

Kind regards,
—*Brian Carpenter* `brian@dxcoms.cern.ch`

Brian,

Thank you very much for your update. I am sure Marshall Rose will be thrilled to hear that his RFC 1006 ("ISO transport services on top of the TCP: Version 3") is now an International Standard. Keep those letters coming!

—*Ole* `ole@interop.com`

Restore Order to the Corporate Network by Reducing Complexity Where It Matters Most

by
Nick Lippis, John Morency and Eric Hindin,
Strategic Networks Consulting, Inc.

Introduction

More and more money is being pumped into the corporate network, yet quality-of-service is no better today than a year ago and prospects for budget relief are nowhere in sight. It's an all-too familiar scenario to many network managers. Is there hope?

The answer is yes. A recent "cost-of-network complexity" study performed by Strategic Networks Consulting, Inc., provides the information that network managers need to formulate effective strategies for restoring order to their networks. The study pinpoints the exact components that make a network more complex and lead to rising costs and reduced quality-of-service. It also provides a way to quantify savings that would accrue as a result of changes to each of the components and compare complexity levels across corporations.

Without this type of information, all the committees, task forces and five-year plans that network managers put in place to restore order are doomed before they even get started. Their efforts are likely to amount to little more than a crap shoot—a committee may be able to identify some of the factors responsible for network problems, but, barring luck, they don't really know the specific areas to concentrate on to produce maximum relief.

The worst may be yet to come for these network managers' companies. Unchecked network complexity sets in motion a series of events that ripples out of the network group into nearly all IT departments and business centers, including sales, marketing, engineering, etc. First, the support effectiveness and buying power of the network group is impacted, then quality-of-service suffers, and, eventually, business units feel the crunch (Figure 1).

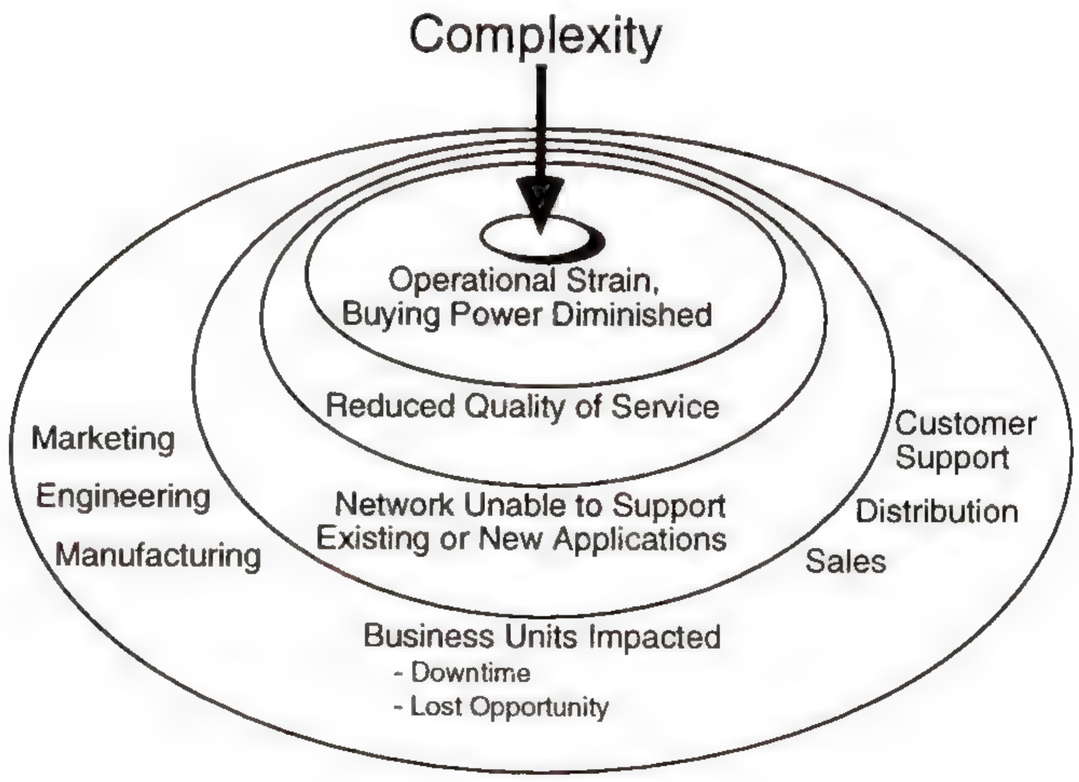


Figure 1: Network Complexity Chain Reaction

The cost-of-network complexity study involved ten large corporations. We determined the individual components responsible for increasing complexity in the study participants' networks and lumped them into categories which we labeled "complexity drivers." A total of seven such drivers were identified. The metric we developed to quantify the effect of the drivers on a corporation's well-being is called "complexity inflation." We call it this because it is analogous to accepted principles of economic inflation.

The benefits a corporation will experience as a result of undertaking this type of study are unmistakable. All the study participants were experiencing at least some of the negative effects described above as a result of the presence of multiple complexity drivers in their networks. In a few cases, network managers had, by going with their instincts, successfully identified general or even specific areas to focus on in order to affect positive change (Figure 2); these managers were nevertheless able to tweak their efforts after reviewing the study results.

"Our next challenge will be to reduce our dependency on legacy protocols like DECnet and AppleTalk."

— *Insurance company*

"The only way to reduce complexity is to reduce the number of vendors and develop a network architecture."

— *Software developer*

"One of the biggest challenges we have in effectively running the network is effective protocol monitoring."

— *Network equipment vendor*

"Nothing drives up complexity and operational budgets as much as large-scale network applications.

— *Pharmaceuticals manufacturing*

Figure 2: User Insight into Network Complexity

For other participants, the benefits were more profound. The study made them realize that they had to adopt radically different approaches to solving their problems. We assume that the experiences of the participants in the study are indicative of that which any network manager can expect if they undertake a cost-of-network complexity study.

Observations

In addition to verifying the premise that a corporation's overall well-being is negatively impacted by increased network complexity, our cost-of-network complexity study yielded a number of other noteworthy observations and conclusions:

- The major cost associated with complexity is the burden it places on support staffs and people budgets, as opposed to other cost areas such as equipment and facilities. As such, a complexity driver, such as number of protocols in a network, is best measured (and will be measured in this article) in terms of its effect on people costs. This finding as to the relative importance of people costs is consistent with the findings of previous cost-of-network ownership studies conducted by Strategic Networks [1].
- The most significant complexity driver (the one that most significantly affects people costs) is network management. The study broke network management down into network management applications and agents, and computed complexity inflation associated with each. Together, these categories easily outdistance the complexity driver with the second most significant affect on costs, the number of protocols installed in a network. Seven out of the ten companies that participated in the study reported a skyrocketing number of network management applications and agents.

continued on next page

Reducing Complexity *(continued)*

- The economic metaphor that explains the affect of increased network complexity carries particular clout with high-level managers when used as a means to win support for programs aimed at reducing complexity. Basically, the economic metaphor says that people resources don't go as far as they might because complexity increases the support resources required to deliver a given quality-of-service. Another way of understanding the metaphor: with economic inflation, there are too many dollars chasing too few goods; with complexity inflation, there are too few network wizards chasing too many network technologies.

- There isn't necessarily a relationship between the number of desktops in a network and the existence of complexity drivers. However, if complexity drivers are present, an increasing number of desktops will act as a multiplier on complexity, thus exacerbating its negative effect.

- There isn't necessarily a relationship between high revenue growth and high complexity. A large bank transitioning from an SNA network to an internet might be plagued by the same complexity levels as a small company implementing a small internet to link NetWare servers.

- The wisdom of reducing the number of equipment suppliers, accomplished by switching to vendors that provide everything from network interface cards to backbone routers, is confirmed by the study—a high number of vendors supplying equipment was the third most important complexity driver (biggest contributor to increased people costs). This speaks highly for the use of, and future success of, "one-stop-shopping" vendors like 3Com or Bay Networks, as opposed to so-called "best-of-breed" suppliers [2].

Complexity inflation

The term *complexity inflation* is all important as a means to understand how increased network complexity negatively impacts a corporation. Complexity inflation attempts to apply known inflation principles in the field of economics to networking. The curved line in Figure 3 shows the accepted notion in economic circles that buying power decreases as inflation increases (pick any point along the curved line and map it to the two axes).

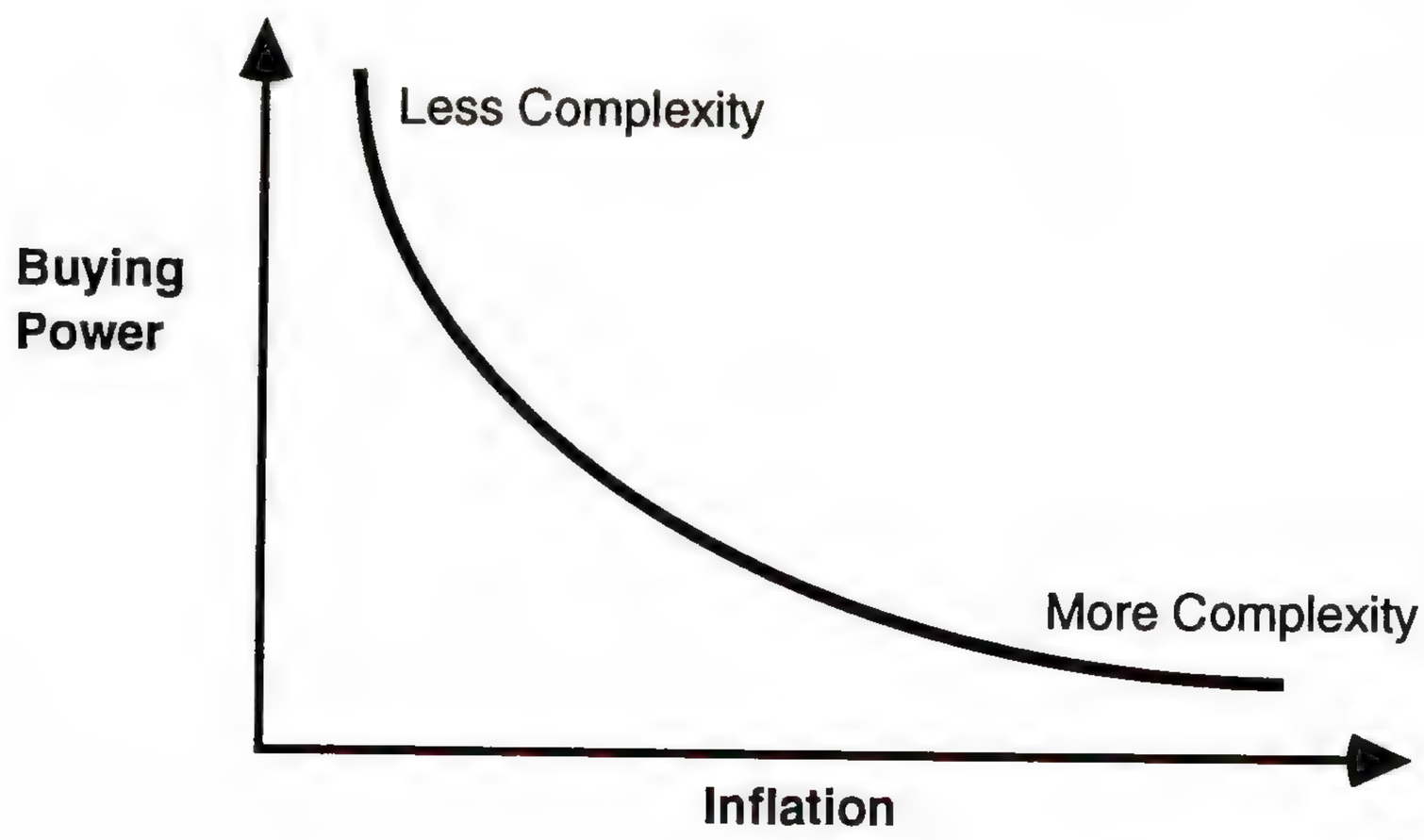


Figure 3: The Inflation/Complexity Metaphor

The economic principles illustrated in Figure 3 can be applied to networking by labeling the two ends of the curved line "less complexity" and "more complexity." Then, Figure 3 shows that, as complexity increases, buying power decreases and network inflation increases, and, as complexity decreases, buying power increases and inflation (*complexity inflation,* as we call it) decreases. Thus, increased complexity acts like inflation on network operation budgets. The more complex a network, the fewer services an operations group can deliver.

Another way to understand this is through the use of the previously-mentioned parallel between the accepted economic notion of "too many dollars chasing too few goods" and the phrase "too much infrastructure chased by too few wizards." The metaphor can be extended by matching accepted economic terms with network terms, as shown in Figure 4. Inflation equates to increased complexity. Buying power can be matched with support expectations, quality-of-life to quality-of-service, inflation drivers to complexity drivers.

Economy		Networks
Inflation	——————>	Complexity Increase
Buying Power	——————>	Support Expectations
Quality of Life	——————>	Quality of Service
Inflation Drivers	——————>	Complexity Drivers

Figure 4: Inflation/Complexity Term Matching

Figures 5 and 6 further illustrate the concept of complexity inflation. Figure 5 illustrates the point that quality-of-service decreases as complexity increases, unless the budget is perpetually increased. The alternative way to maintain the quality-of-service level represented by the straight line going across the graph is to decrease complexity. This fiigure also illustrates the point that quality-of-service eventually deteriorates as complexity increases regardless of how big a company or its budget is.

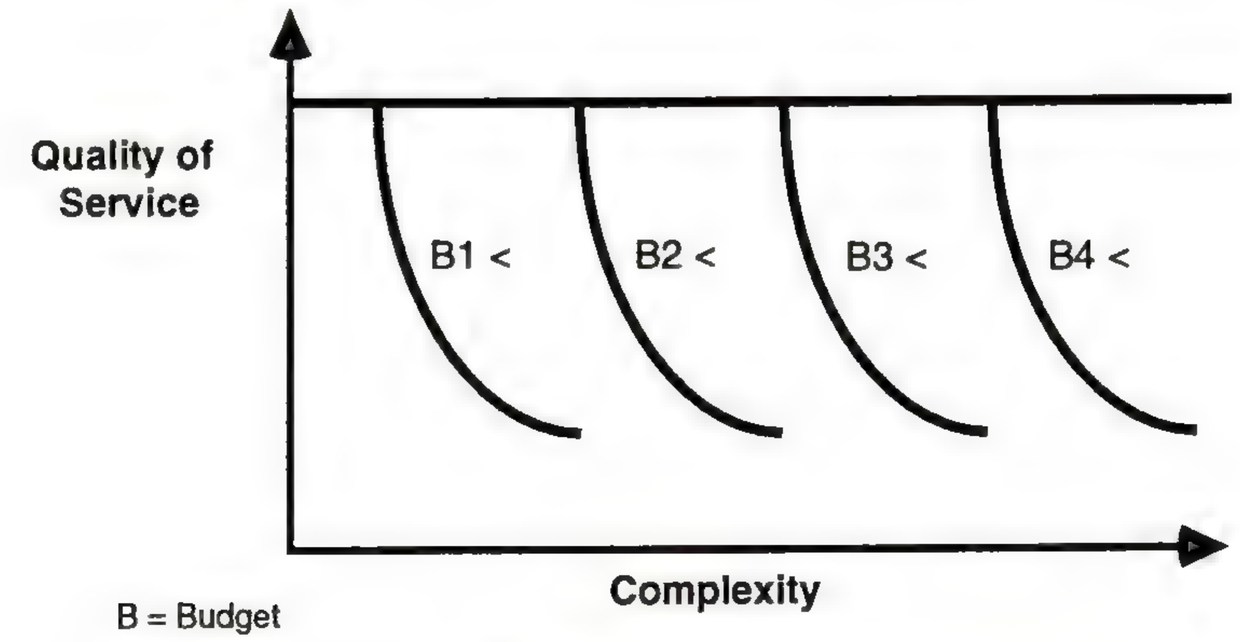


Figure 5: Complexity's impact on Budget

Figure 6 illustrates complexity inflation in a different way. It shows the impact of increased network complexity on budgets as a change is made that increases network complexity. When the change is made, budgets and resources are strained to accommodate the change. The change in day-to-day operations required to implement the new service is complexity inflation. The initial cost to implement the new service is subtracted out of the operations group. This subtraction of cost and/or resources is a direct effect of absorbing complexity inflation.

continued on next page

Reducing Complexity *(continued)*

After the change is made, the network again enters a steady state mode of operation, albeit at a new, higher cost mode of operation. This equates to diminished buying power.

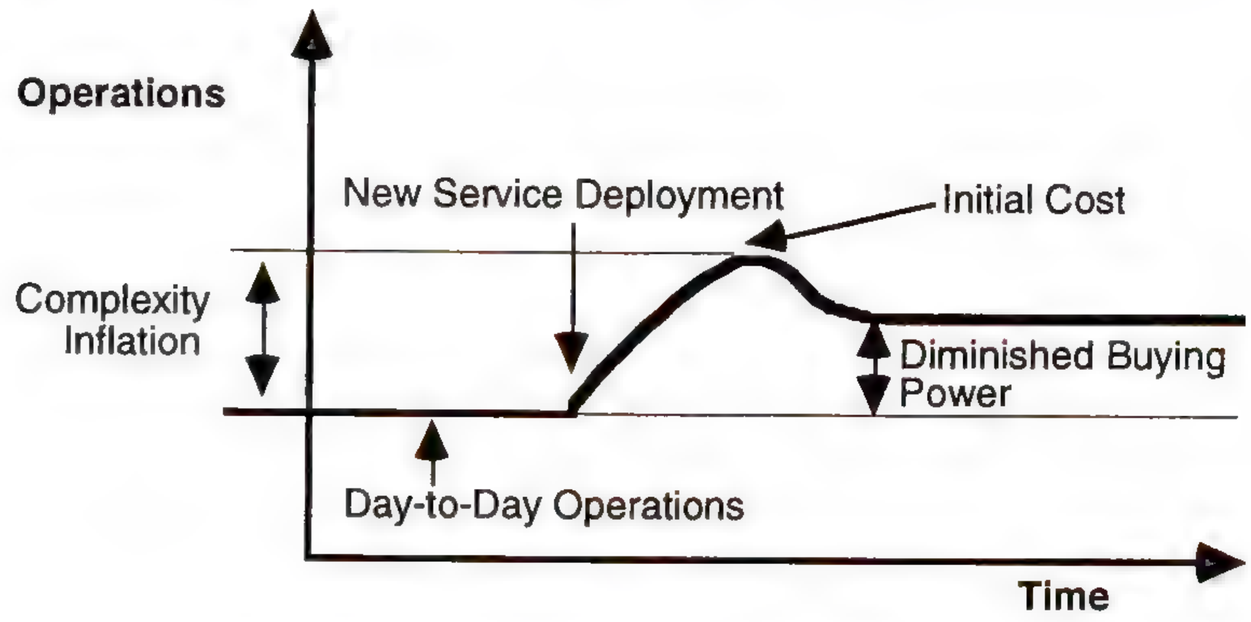


Figure 6: Complexity Inflation at Work

With this understanding of what's meant by complexity and complexity inflation, it is possible now to discuss exactly how complexity inflation can be calculated.

Measuring costs

Complexity inflation is primarily a measure of how increased complexity affects people costs. Other cost areas, such as capital equipment and facilities budgets are also impacted, but this impact is less direct: it comes as a result of complexity's affect on people costs. Complexity's tendency to primarily affect people costs is particularly troublesome, since people costs are the biggest cost item associated with running most networks.

The specific cost characteristics of today's typical network were reported in the November, 1993 issue of *The Internetwork Advisor*. [1] As such, they are only summarized here.

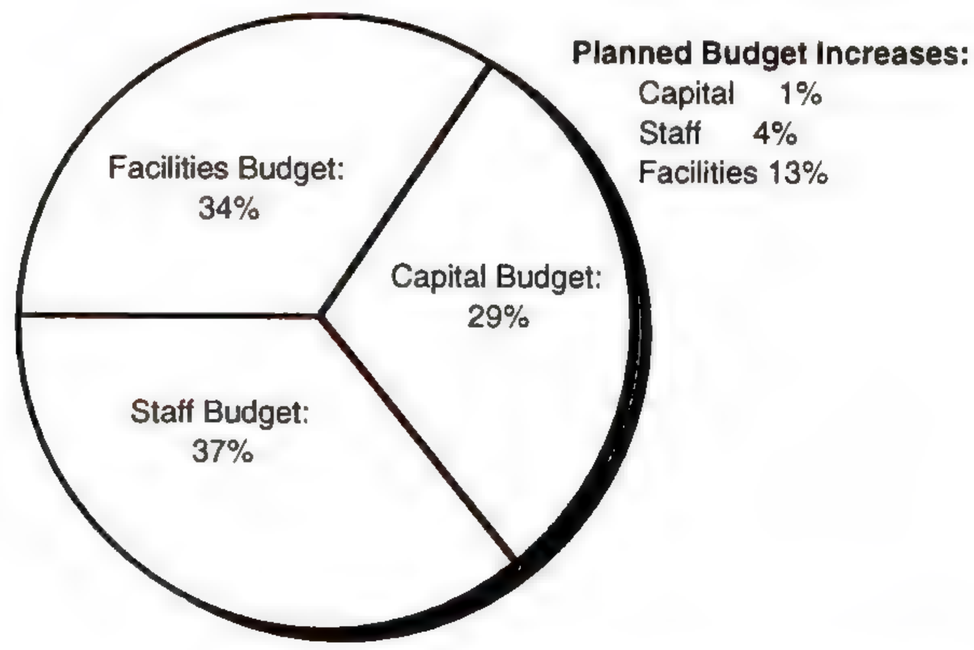


Figure 7: The Cost of Network Ownership

As shown in Figure 7, the average capital equipment budget reported by the ten cost-of-network complexity study participants represented 29 percent of total network budget, while people and facilities budgets consumed 37 and 34 percent, respectively. The relative position of each budget remains unchanged from our original cost-of-ownership study, though this latest study shows that capital and facilities budgets are now somewhat closer in terms of size. The previous breakdown placed capital and facilities at 26 and 32 percent, respectively, with staff budgets at 37 percent. The change is probably due only to changes in sample size, rather than any fundamental shift in the industry.

Other metrics that quantify network ownership costs include cost per networked desktop, staff cost per networked desktop and network desktops per support person. The ten cost-of-network complexity study participants had an average annual cost per networked desktop of $2,045; the annual staff cost per networked desktop was $250 and there was 356 networked desktops per support person. Again, these results are consistent with the original cost of network ownership study.

Complexity drivers

Quantifying the specific impact on people costs associated with increased complexity (thus allowing complexity inflation to be calculated) requires the identification of the specific elements contributing to complexity. These elements are grouped into categories called *complexity drivers*. Since people are the biggest cost item associated with networks, complexity drivers can thus be defined as areas of the network that consume people resources.

Complexity Driver	Insurance Company			Study Average
	Today	Two Years	Rank	Today/Two Years
Management Applications	10	7	2	4/6
Management Agents	0	3	2	5/10
Protocols	14	14	1	7/6
Platforms	6	8	1	5/6
Equipment Types	35	35	1	9/9
Network Applications	215	230	1	34/43
Number of Vendors	14	14	1	9/9

Figure 8: How Much Complexity?

Strategic Networks identified seven major complexity drivers for the ten users who participated in the cost-of-network complexity study. These are shown in column one of Figure 8. We believe that these drivers characterize most all networks installed in the U.S. and throughout the world.

Most of the categories are self-explanatory. The platforms category refers specifically to server operating systems such as NetWare or Windows NT. Equipment types refers to bridges, routers, hubs, etc. The vendors category reflects the number of suppliers.

The rest of Figure 8 illustrates complexity levels reported by the study participants. Specifically, it shows the number of installed components within each complexity driver category today and two years from now. The study averages shown in the last column in the table do not seem to paint all that bleak a picture. This doesn't mean, however, that there aren't plenty of participants in dire straits. The columns labeled "Insurance Company" illustrate this point. They show one of the more worse-off companies that participated in the study and its rank in each category compared to the other study participants.

continued on next page

Reducing Complexity *(continued)*

Budget impact

Knowing that complexity drivers exist in a network does little to help efforts to calculate complexity inflation. What's needed is an assessment of the exact effect of each complexity driver on people costs.

The effect on support staff time of each complexity driver was determined via an exercise with study participants where all tasks associated with changes to a complexity driver were written down, along with required completion times. These tasks and associated time periods were classified as either "initial cost" items or "steady state cost" items, depending on whether they were one-time tasks associated with installation or recurring items associated with maintenance.

For example, for the protocols complexity driver, initial costs included the total amount of time needed to learn about a new protocol and install it in routers and desktops. Steady state costs include time to tune and reconfigure software in response to moves, adds and changes, and changes to the network topology.

Complexity Driver	Initial Cost	Steady State Cost
Management Applications	2.2	1.2
Management Agents	2.2	1.7
Protocols	4.2	2.0
Platforms	2.8	1.2
Equipment Types	2.3	0.8
Network Applications	2.1	2.5
Number of Vendors	4.0	0.9

Figure 9: Complexity Driver Impact

Figure 9 shows the initial cost and steady state cost, measured in terms of person months, associated with each complexity driver. The study broke network management into two complexity drivers, management applications and agents. Learning and installing a new management application requires 2.2 months (the initial cost) and carries a steady state cost of 1.2 months. As mentioned, network management had the biggest impact on complexity.

Protocols proved to have the second biggest impact on complexity, consuming an average of 4.2 months to learn and 2.0 months per year to maintain, while the addition of a new platform finished third, consuming 2.8 months to learn and install and 1.2 months per year to maintain.

When using the initial and steady state cost ratings in Figure 9 to determine what areas within a network to focus the most attention on to reduce complexity, it also helps to consider the relative ranking of different drivers in the initial cost and steady state categories. In other words, a driver that carries a high initial cost might not necessarily have a high steady state impact.

Either attribute may be more desirable or more or less applicable based on factors such as support personnel skill level and expertise in one complexity driver versus another.

Figure 10 lists the seven complexity drivers in their order of importance according to both the initial costs and steady state attributes. The most telling conclusion to be drawn from Figure 10 concerns network applications. While this complexity driver has the least impact when evaluated in terms of initial costs, it has the largest effect on steady state costs. Management applications was the only other complexity driver with a greater steady state cost than initial cost. The differences between complexity driver's initial and steady state costs not only aid in planning how to reduce network complexity, they point out important product purchase criteria, especially in the area of network management.

Ranking	Initial Cost	Steady State Cost
1	Protocols	Network Applications
2	Number of Vendors	Protocols
3	Platforms	Management Agents
4	Equipment Type	Platforms
5	Management Applications	Number of Vendors
6	Management Agents	Equipment Type
7	Network Applications	Management Agents

Figure 10: Complexity Driver Dynamics

Complexity inflation math

The complexity drivers along with their associated initial cost and steady state metrics form the basis for measuring complexity inflation. To calculate complexity inflation for a given user over, say, a three-year period, the year-to-year change for each of the complexity drivers must first be determined. For example, if the complexity driver is protocols, the number of protocols present in the first year would first be determined, then the number of protocols to be added or subtracted during the next year would be determined. A company with four protocols during one year and five the next would have a change of plus one. The amount of change for each of the complexity drivers is then multiplied by the initial cost value for the complexity drivers (listed in Figure 9) and the results are added together.

This exercise is then repeated to reflect change occurring from year two to year three. The only difference is that the change values are multiplied by the steady state cost value for each complexity driver, rather than the initial cost values.

The year one to year two results are then added to the year two to year three results. This number is then divided by 80% of the total number of staff people available multiplied by the number of months in the study. 80% is used to allow for overhead. The final number is the complexity index. The next section defines acceptable and non-acceptable indexes.

Breaking points

The complexity inflation metric comes in handy in a number of ways. First, it can be used to determine whether new services can be deployed and new projects deployed without adversely affecting a corporation's overall well-being.

continued on next page

Reducing Complexity *(continued)*

Based on the data provided by the ten participants in the cost-of-network complexity study, it is possible to determine acceptable and non-acceptable levels of complexity inflation for the typical corporate network.

Complexity inflation less than five percent does not appear to significantly impact corporate well-being. Support staffs are able to handle this growth without budget increases or quality-of-service shortfalls.

However, a breaking point occurs when complexity inflation exceeds five percent and the number of desktops in the network is growing (Figure 11). When this scenario and any scenario where even greater complexity inflation and growth occurs, the chain reaction described in Figure 1 is set in motion. Specifically, the combination of six percent complexity inflation and the addition of 500 desktops to the corporate network results in a one-to-three person shortfall.

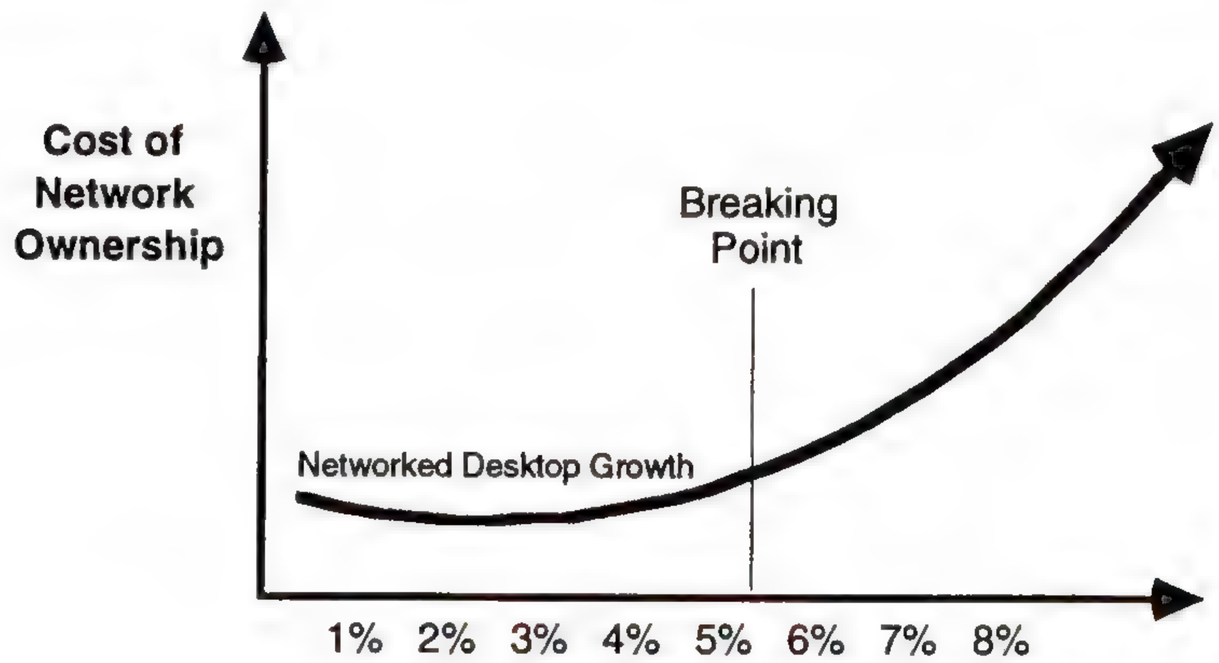


Figure 11: Complexity Inflation Breaking Point

The average number of desktops per staff person for the study is 1:320. The breaking point for corporations with smaller ratios may be somewhat higher than the six percent/500 desktop breaking point observed for the study participants. To confirm that these numbers are accurate, Strategic Networks analyzed 42 additional firms listed in its database including those that participated in past cost-of-network ownership studies. A look at these firms' budgets and support staff increases made during the past few years, or planned in the coming years, confirmed the integrity of the data.

Another way of using the complexity inflation metric is to predict the viability of any single project. For example, if an ATM workgroup is to be installed, at least four complexity drivers, protocols, equipment types, number of vendors and management agents would be added, requiring a total of 13.5 person months of staff. If this addition moves a firm past the five percent complexity inflation/additional desktops breaking point, then perhaps reconsideration of the ATM workgroup is in order.

What can be done?

If overall complexity inflation is too high or it is determined that a single project will move the corporation over the breaking point, several short- and long-term coping strategies can be adopted.

The first and most obvious strategy is to postpone planned expansion or specific projects. Of course, this is not desirable, especially if money for the project has already been allocated.

A second strategy is to bring in temporary staff or outsource certain aspects of day-to-day operations. This approach will allow planned expansion and new projects to proceed, but could end up being costly.

The long-term solution is to eliminate some complexity drivers from the network. Essentially, this amounts to trading off existing support requirements in favor of new ones that planned expansion and projects carry.

The particular complexity drivers to drop requires a personalized decision, since every corporation has different priorities. However, an analysis of the successes and failures experienced by the cost-of-network complexity study participants does yield some general guidelines.

It seems apparent from the study that it is best to tradeoff complexity drivers with high initial cost in favor of those with a high steady state cost. For example, the total number of vendors and protocols complexity drivers should be reduced, while the network and management applications drivers should be increased. Both these high steady state cost complexity drivers yield a high return on the investment they require, either in terms of advancing the corporation's ability to be competitive (network applications) or by potentially reducing the negative effect inflicted by other complexity drivers (network management).

The idea that one complexity driver can negatively or positively impact another is not reflected by the complexity inflation metric. The metric speaks only to the negative or positive affect associated specifically with supporting the complexity driver. The relationships between complexity drivers should be taken into account when planning how best to reduce complexity. However, the potential positive effect associated with adding complexity should never be overestimated. A network management vendor, for example, may claim that its program will completely or partially mitigate other complexity drivers such as number of protocols or platforms, but Strategic Networks has never run into anybody who had actually succeeded in reducing complexity by implementing anything new.

Study results

About half of the cost-of-network complexity study participants had complexity inflation below the five percent/desktop growth breaking point. The study low was a petroleum company with negative eight complexity inflation. The study high was a diversified services firm at six percent.

Those firms keeping complexity at bay all have adopted the same general strategy: they are increasing complexity drivers as little as possible in existing networks while balancing new service deployment against a bounded staff size. The one exception is a software development firm. It appears able to get away with adding a huge number of new services. This is probably due to the aggressive complexity reduction strategy it has adopted. It may also have a leg up on other users who participated in the study because its typical employee is network savvy and makes fewer demands on full-time support staff.

The specific components of the software company's complexity reduction strategy as well as highlights of other study participants' efforts (or lack of effort) is shown in Figure 12. The figure also shows where each corporation is in terms of number of desktops and staff cost per desktop, so that any network manager performing a cost of network complexity study on his/her own can make the most meaningful comparisons.

continued on next page

Reducing Complexity *(continued)*

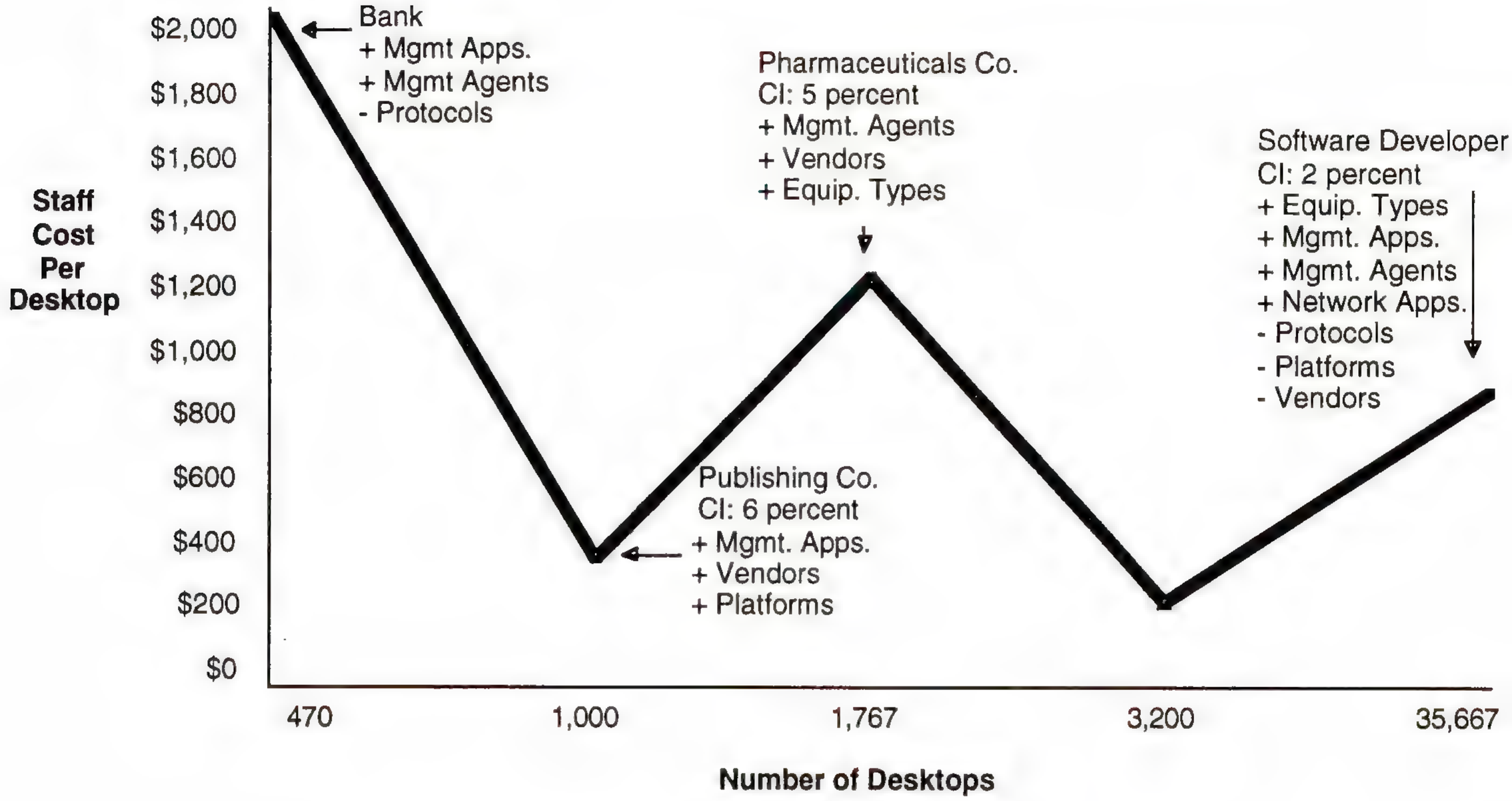


Figure 12: Complexity Snapshots

References

[1] "Internetwork Cost of Ownership Hits the WAN Brick Wall," *The Internetwork Advisor,* November, 1993).

[2] "SynOptics/Wellfleet: Thumbs Up for a Merger that Will Benefit the Entire Network Industry," *The Internetwork Advisor,* July, 1994).

NICHOLAS J. LIPPIS III is president and founder of Strategic Networks Consulting, Inc., a full-service computer network consultancy. He is a world-renowned authority on advanced enterprise internetworking architectures, implementations and management. Mr. Lippis has consulted to numerous Fortune 1000 firms on the topic of enterprise network strategy, including Barclays Bank, Schering-Plough Research Institute, Hughes and Liberty Mutual. He is a contributing editor and monthly columnist in *Data Communications* magazine and a senior member of the NetWorld+Interop program committee. E-mail: `lippis@snci.com`

JOHN P. MORENCY was previously a principal consultant with Strategic Networks. He has over twenty years of industry experience as a product developer, network manager and industry consultant. He is the author of the industry's first Buyers Guide for ATM, the co-inventor of the internetworking cost of ownership model and principal for a major study on industry-wide switched internetworking trends. He is a regular columnist for *Network World* and a member of the program planning committee for Networks Expo. Mr. Morency holds a bachelor's degree in mathematics and computer science from the University of New Hampshire.

ERIC M. HINDIN is a senior consultant with Strategic Networks. His expertise centers around the design, features and functionality of all types of internetworking products and the use of those products in enterprise networks. Mr. Hindin consults with Strategic Networks' clients and writes the company's monthly newsletter, *The Internetwork Advisor.* He is frequently quoted in the trade press, and speaks regularly at industry trade shows and university seminars. Mr. Hindin has been a columnist for *Data Communications* Magazine and PC Week, and has also been published in *Communications Week* and *UNIX World.* E-mail: `hindin@snci.com`

DANTE awards EuropaNET Continuation Contract to British Telecom

Introduction

Following completion of a public invitation to tender, British Telecom (BT) has been awarded a contract to provide the continuation of the current EuropaNET backbone service. EuropaNET is the pan-European network that interconnects the academic and research networks in Europe and connects them to the global Internet. EuropaNET represents a major part of the Internet in Europe. The network is managed by DANTE, a company set up in 1993 by the national research networks in Europe.

Under the contract the pan-European backbone will be outsourced to BT, which includes backbone planning (in cooperation with DANTE) and implementation, the provision of switching equipment, circuits, and help desk facilities. The new contract provides for an IP (Internet Protocol) service at access speeds up to 8Mbps.

The network will be open to commercial user organisations. DANTE and BT will sign an agreement under which commercial usage is handled by BT to the mutual benefit of DANTE's customers and BT.

34 Mbps lines

The next step will be to establish a network based on 34Mbps lines. As these are not available yet and there was no guarantee they would be by the time the current EMPB contract expires (October 1995) an arrangement for a continuation of EuropaNET was mandatory.

Dai Davies, general manager of DANTE said: "The new contract gives us the space we need to prepare for the implementation of the first 34Mbps connections. The choice for BT is based on solid technical and commercial grounds. It was a close finish but we are convinced this outcome is in the best interest of our customers. Strong points in the BT offer were their advanced plans for a Europe-wide IP service as well as their global reach via their linkup with MCI."

Transition

DANTE will use the coming months to prepare for a smooth transition to ensure that EuropaNET customers will enjoy an uninterrupted high quality service.

More information

For more information please contact:

Josefien Bersee
External Relations Manager
DANTE
Lockton House
Clarendon Road
Cambridge CB2 2BH
United Kingdom
Phone: +44 1223 302 992
E-mail: J.Bersee@dante.org.uk

For further reading

[1] Bersee, J., "Profile: DANTE and EuropaNET," *ConneXions,* Volume 8, No. 6, June 1994.

[2] *ConneXions,* Volume 7, No. 5, May 1993, "Special Issue: Focus on Europe."

[3] Stockman, B., "EBONE, The European Internet Backbone," *ConneXions,* Volume 7, No. 5, May 1993.

[4] "Réseaux Associés pour la Recherche Européenne (RARE)," *ConneXions,* Volume 6, No. 1, January 1992.

Book Review

ISDN and Its Application to LAN Interconnection, by Dervis Z. Deniz, 254 pages, McGraw Hill, 1994. ISBN 0 07 707883-7.

This is a very technically detailed book about the use of narrow band ISDN (both basic rate and primary rate) for LAN interconnection. The *Integrated Services Digital Network* access is becoming very widespread in Europe and Japan, for a variety of reasons.

Background

Firstly, the deregulation of telephone companies has moved a lot more slowly, so that leased line charges have remained inordinately high in most of Europe. This has made the use of switched services such as X.25 and ISDN attractive from a tariff point of view for many businesses with relatively intermittent traffic. Secondly, the geographic scale of Europe has meant that the costs of providing digital access (at least at 64Kbps or 128Kbps or thereabouts) in the subscriber loop for the majority of each national population is often bearable (no expensive line drivers etc). Finally, the Internet was slower to take off in Europe (although it has now done so with a vengeance), so alternative ways of dealing with communications costs such as efficient statistical multiplexing/line cost sharing were not available.

International ISDN access between European countries has been reported to work very well, at least for calls that carry voice or data (ISDN H.261 based videophone usage is less successful, although in this reviewers experience, calls between several countries have succeeded).

The ISDN standards define a whole bunch of things. Primarily, these cover the structure of channels that are used for signaling, and those used for data. Recent advances have meant that "superchannels" or groups of the 64Kbps "B" channels can be accessed together. This requires a special unit called a "Bonding" box that deals with the fact that the separate 64Kbps B channels may be (and often actually are) routed differently through the digital phone exchanges and experience different delays. Basically, bonders take each channel and run an RTT estimator, and then introduce a buffer for all the faster route channels so that the bits on the slower channels can catch up (analogous to the TCP RTT estimator, or the audio adaptive playout used by LBL's Internet multicast audio tool, *vat*).

Of course, all these pieces of technology are all very well if what you want is a *circuit.* But the world of computer networks is largely based around *packets,* and around IP packets (and quite a few Novell packets, I suppose :–). So a box that interconnects sites (typically LANs at sites) needs to worry about the fact that most applications send packets as and when they like (well, modulo a few interferences from TCP or SPP or some transport protocols flow control and error recovery schemes operating, of course).

That is where this book comes in. You need to decide when you want channels, how many you need, where to, and when you've finished with them.

Organization

The book is based on doctoral work done by the author while at UCL, and includes some pretty detailed simulations of traffic, as well as being based on real world experience with ISDN. It is written in 10 chapters. The first three introduce ISDN, superchannels and LAN interconnection.

Chapters 4, 5, 6 and 7 discuss the different approaches to managing and using superchannels. The last 3 chapters discuss queue and rate based channel management algorithms.

Some of the ideas here would be worth a look for some of the Broadband ISDN researchers looking at how to provide UBR and ABR services in ATM networks, since the problems are closely related.

ISDN is here to stay

The problem of transparently providing interconnection of datagram based networks using circuit based networks was one that we thought might have gone away with the emergence of IP and domination over X.25. However, ISDN in basic, primary and broadband services is very much here to stay (for at least a decade) since this provides the Public Network Operators (PTTs or telcos) with several useful capabilities that we will not persuade them easily to part with:

- Trunk Bandwidth Management
- Dealing with legacy networks (the phone system)
- Simple charging

Readers

This book will be of interest to engineers and network designers looking at these thorny interworking problems.

—*Jon Crowcroft, University College London*
J.Crowcroft@cs.ucl.ac.uk

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use it for letters to the Editor, requests for the index of back issues, questions about particular articles etc.:

ConneXions—The Interoperability Report
303 Vintage Park Drive
Suite 201
Foster City
California 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: connexions@interop.com
URL: http://www.interop.com

Subscription information

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 502-493-3217 outside the USA. This is the number for our subscription agency, the Cobb Group. Their fax number is +1 502-491-8050. The mailing address for subscription payments is: P.O. Box 35840, Louisville, KY 40232–9496.

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1